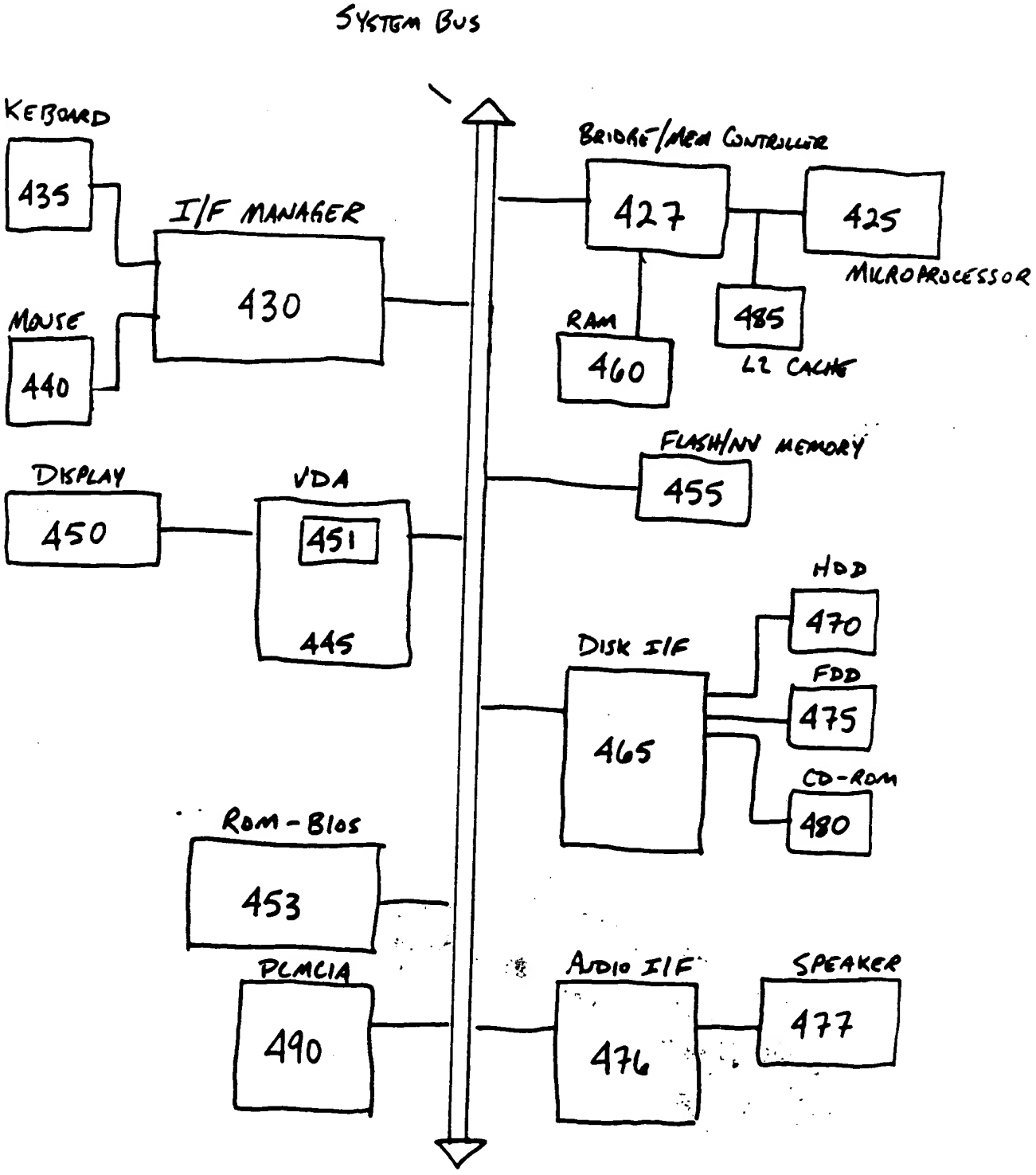

System Bus
Keboard
435
I/F Manager
430
Mouse
440
Bridge/Mem Controller
427
425
Microprocessor
485
L2 Cache
RAM
460
Flash/NV Memory
455
Display
450
VDA
451
445
HDD
470
Disk IF
FDD
475
465
CD-ROM
480
Rom-Bios
453
PCMCIA
490
Audio IF
476
Speaker
477

FIG 1

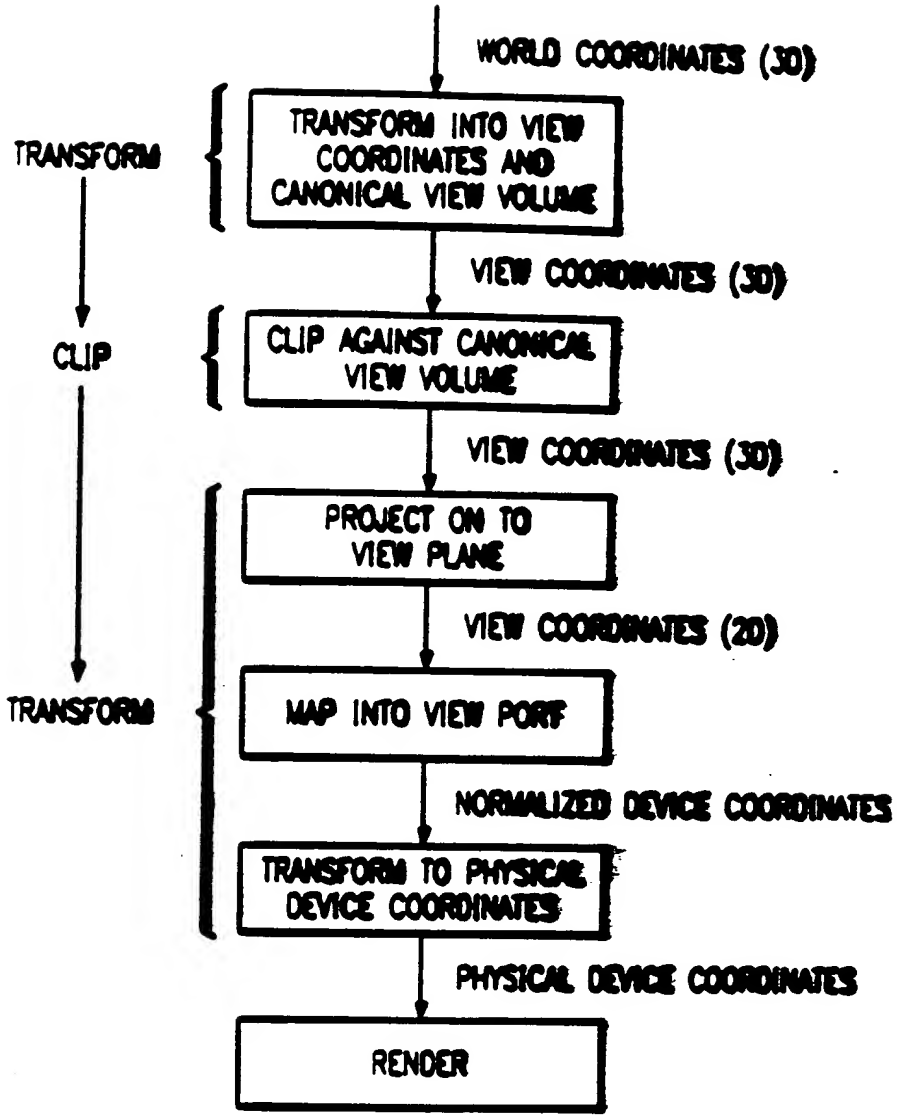

WORLD COORDINATES (3D)
TRANSFORM
TRANSFORM INTO VIEW COORDINATES AND CANONICAL VIEW VOLUME
VIEW COORDINATES (3D)
CLIP
CLIP AGAINST CANONICAL VIEW VOLUME
VIEW COORDINATES (3D)
PROJECT ON TO VIEW PLANE
VIEW COORDINATES (2D)
TRANSFORM
MAP INTO VIEW PORT
NORMALIZED DEVICE COORDINATES
TRANSFORM TO PHYSICAL DEVICE COORDINATES
PHYSICAL DEVICE COORDINATES
RENDER

FIG. 2

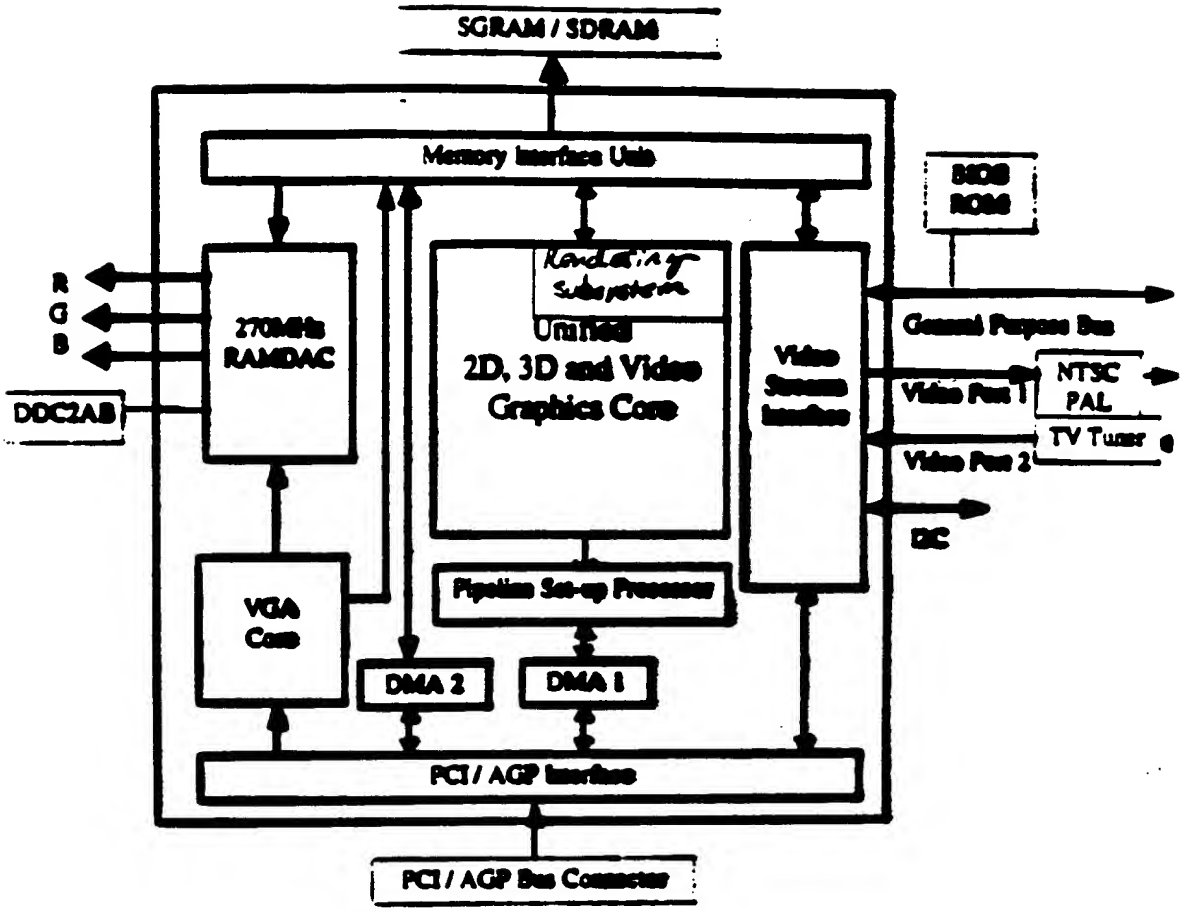

SGRAM / SDRAM
Memory Interface Unit
R
G
B
270MHz
RAMDAC
DDC2AB
Rendering
Subsystem
Unified
2D, 3D and Video
Graphics Core
Video
Stream
Interface
BIOS
ROM
General Purpose Bus
Video Port 1
NTSC
PAL
TV Tuner
Video Port 2
I2C
VGA
Core
Pipeline Set-up Processor
DMA 2
DMA 1
PCI / AGP Interface
PCI / AGP Bus Connector

Figure 3

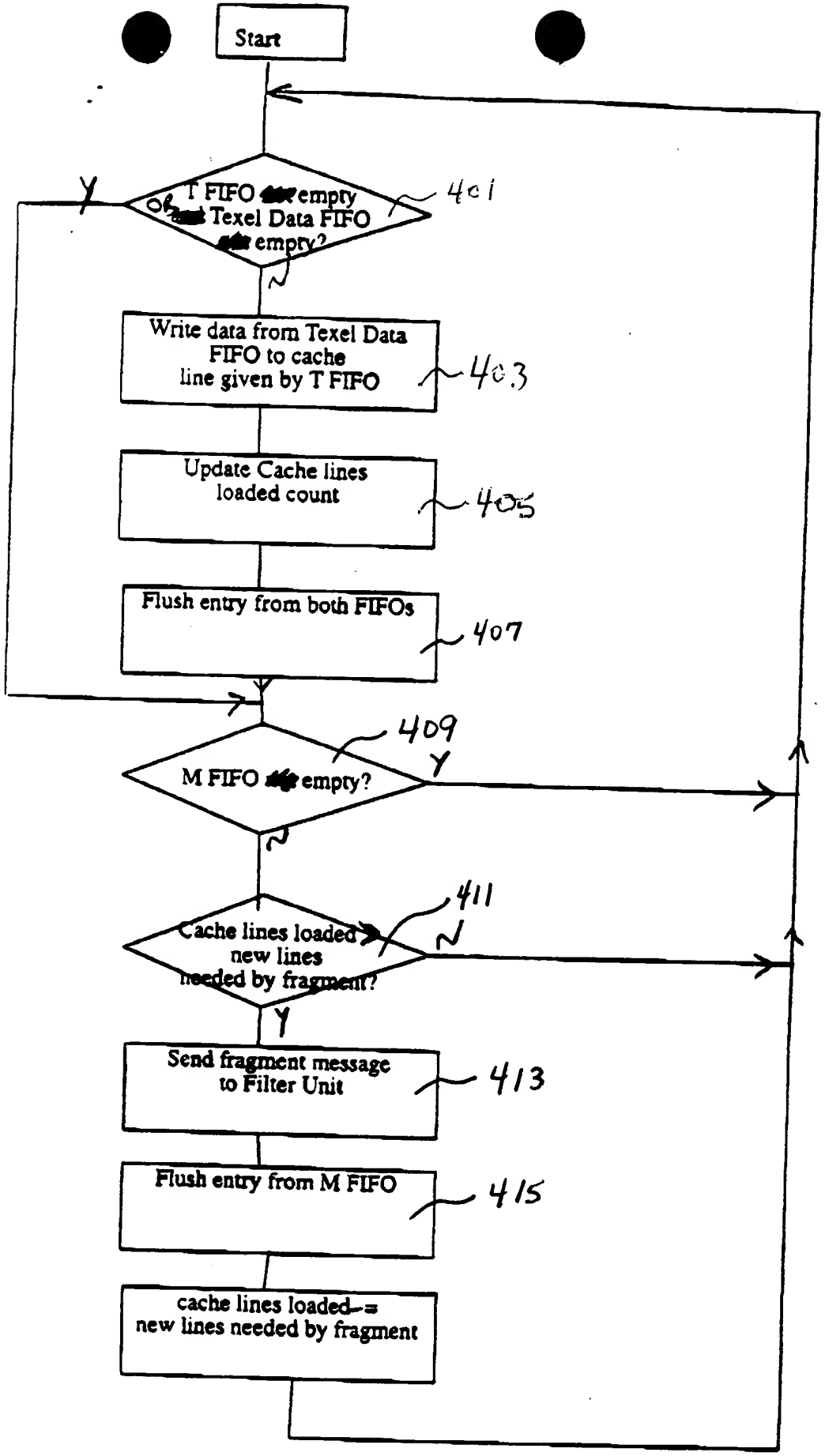

Start
T FIFO empty
Texel Data FIFO
empty?
401
Y
N
Write data from Texel Data FIFO to cache line given by T FIFO
403
Update Cache lines loaded count
405
Flush entry from both FIFOs
407
M FIFO empty?
409
Y
N
Cache lines loaded = new lines needed by fragment?
411
N
Y
Send fragment message to Filter Unit
413
Flush entry from M FIFO
415
cache lines loaded += new lines needed by fragment

FIG. 4A

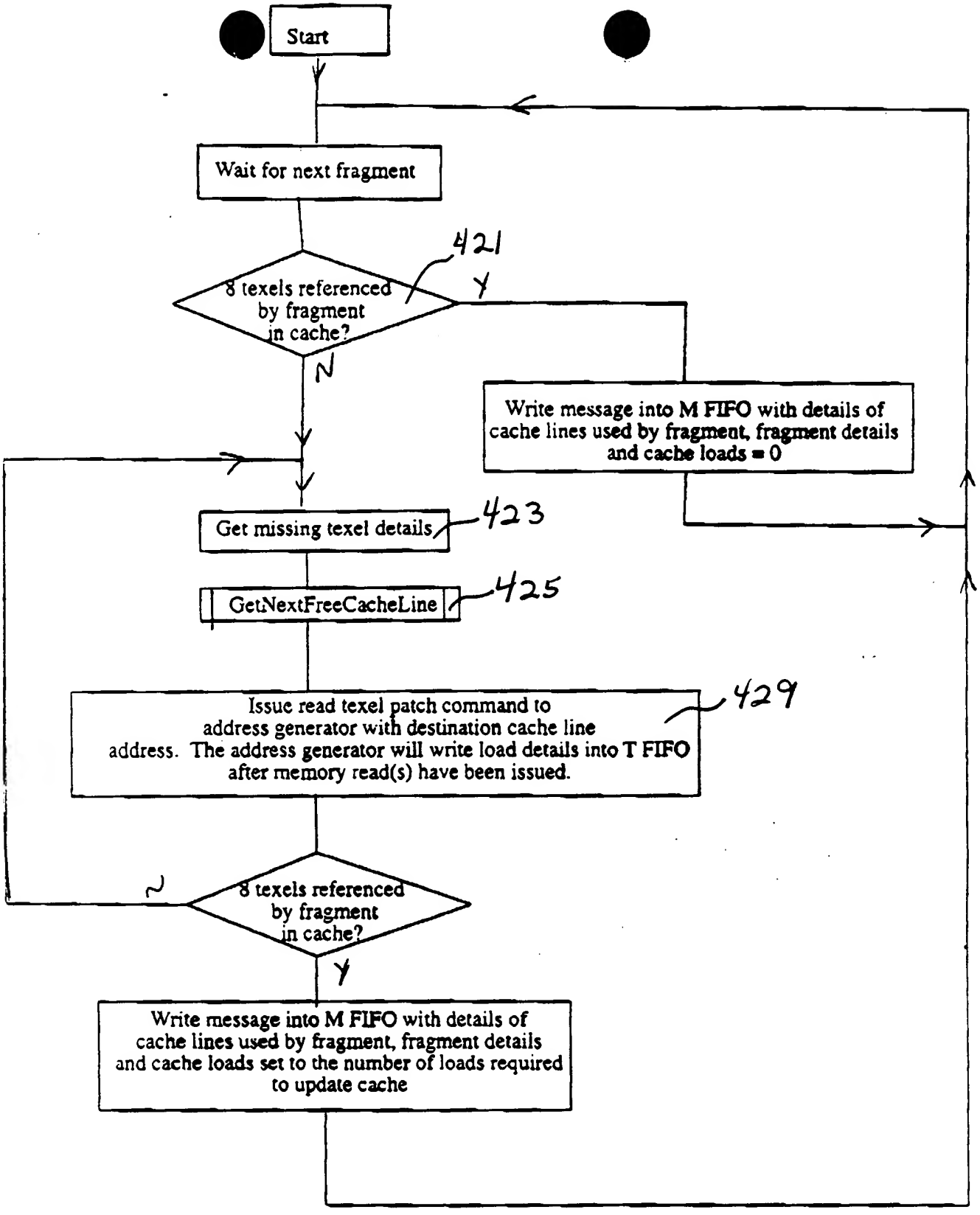

Start
Wait for next fragment
8 texels referenced by fragment in cache?
421
Y
N
Write message into M FIFO with details of cache lines used by fragment, fragment details and cache loads = 0
Get missing texel details
423
GetNextFreeCacheLine
425
Issue read texel patch command to address generator with destination cache line address. The address generator will write load details into T FIFO after memory read(s) have been issued.
429
8 texels referenced by fragment in cache?
N
Y
Write message into M FIFO with details of cache lines used by fragment, fragment details and cache loads set to the number of loads required to update cache

FIG 4B

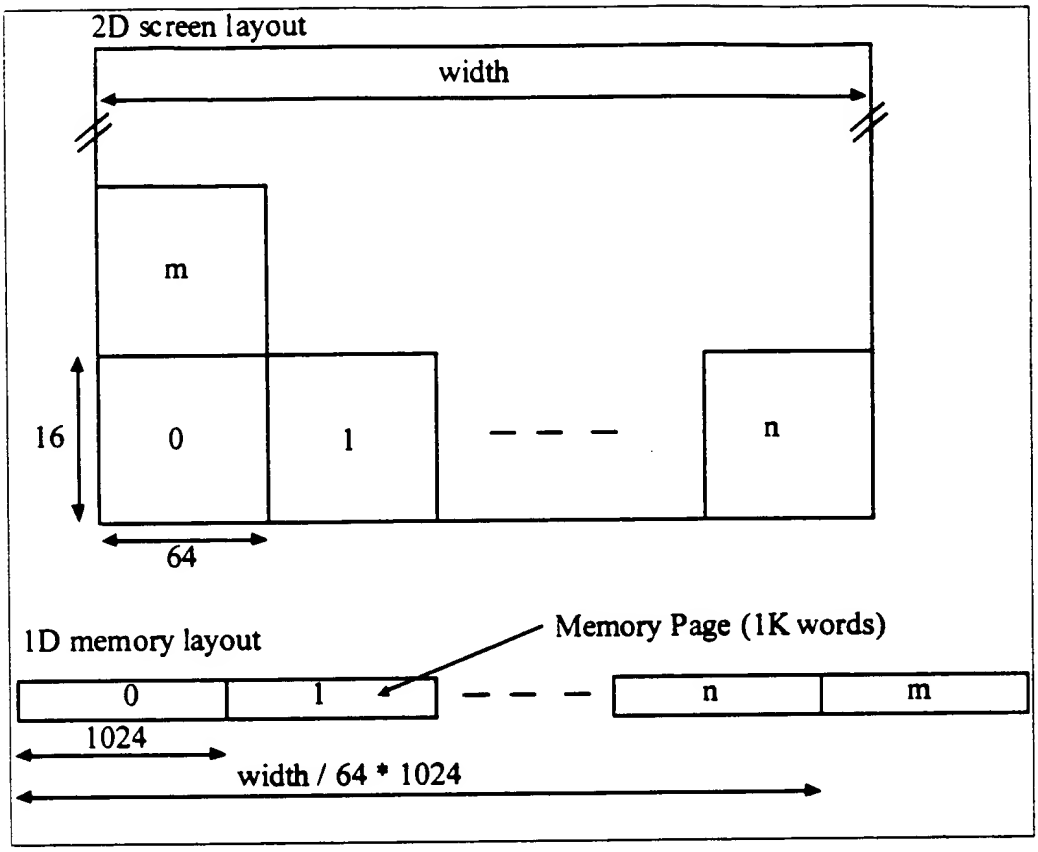
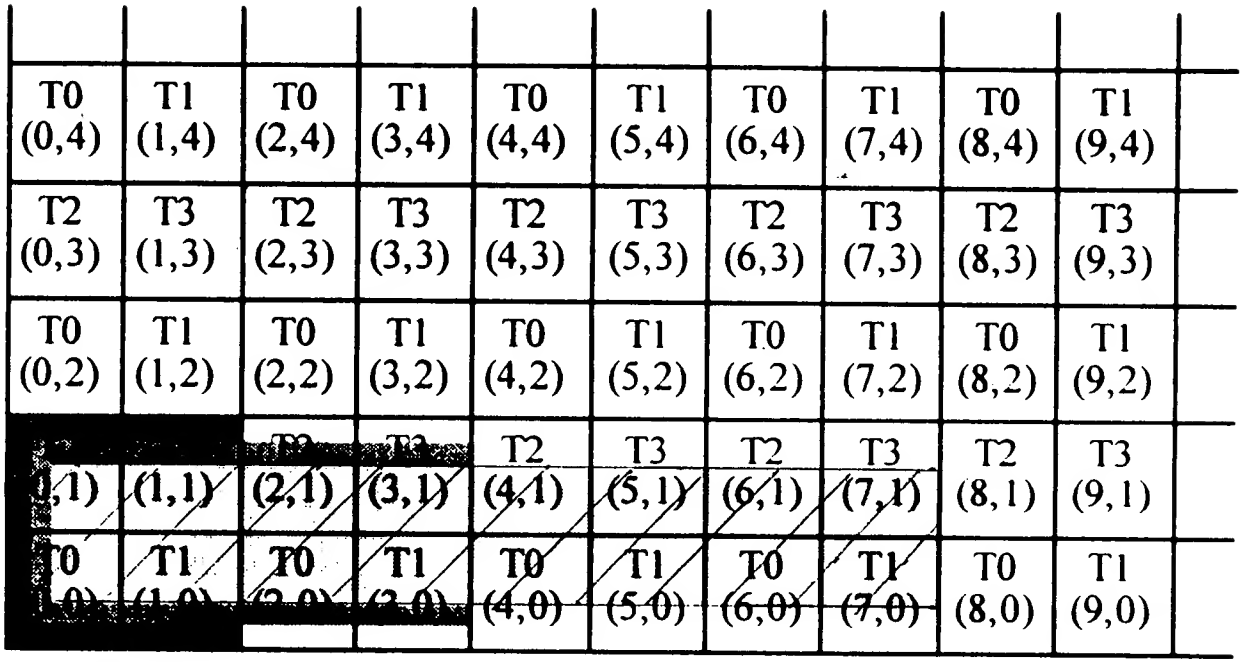

2D screen layout
width
m
16
0
1
n
64
1D memory layout
Memory Page (1K words)
0
1
n
m
1024
width / 64 * 1024

FIG. 5

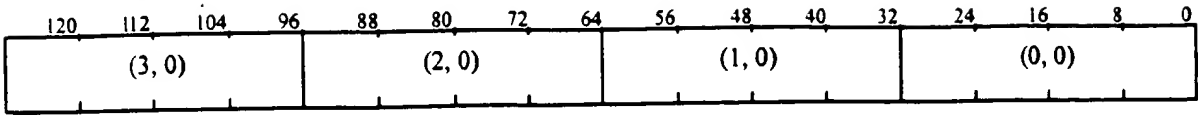


■ 32 bit texels in memory word
■ 16 bit texels in memory word
▨ 8 bit texels in memory word

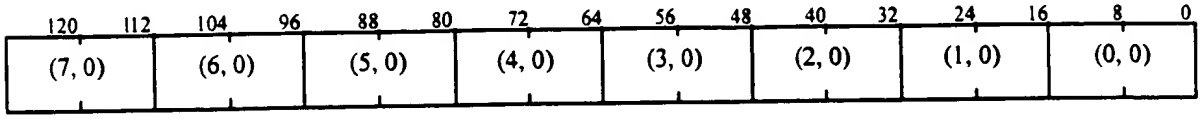
FIG. 6

Linear or Patch64 Memory Layouts

32 bits per texel



16 bits per texel



8 bits per texel

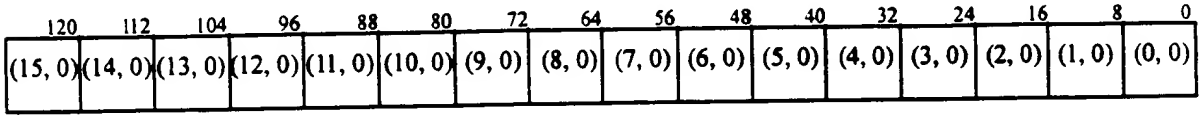
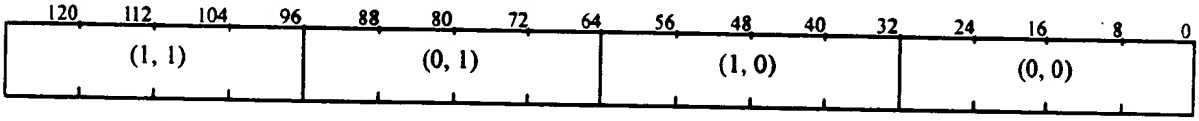


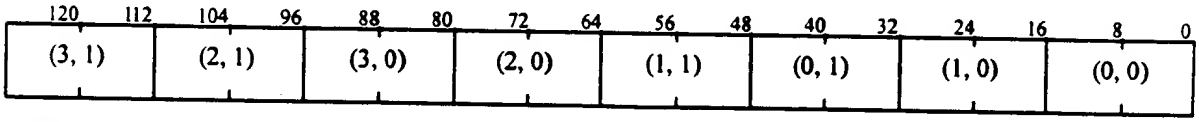
FIG. 7A

Patch32_2 or Patch2 Memory Layouts

32 bits per texel



16 bits per texel



8 bits per texel

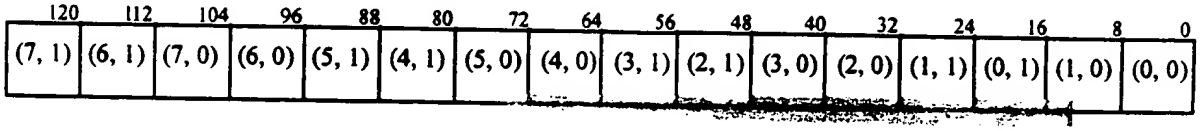


FIG. 7B

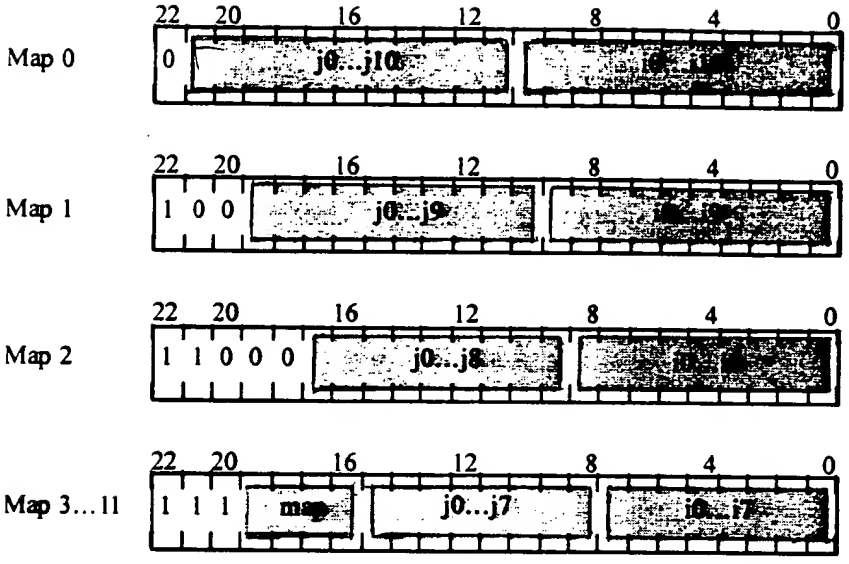

22 20 16 12 8 4 0
Map 0
0
j0...j10
Map 1
1 0 0
j0...j9
Map 2
1 1 0 0 0
j0...j8
Map 3...11
1 1 1
j0...j7

FIG. 8

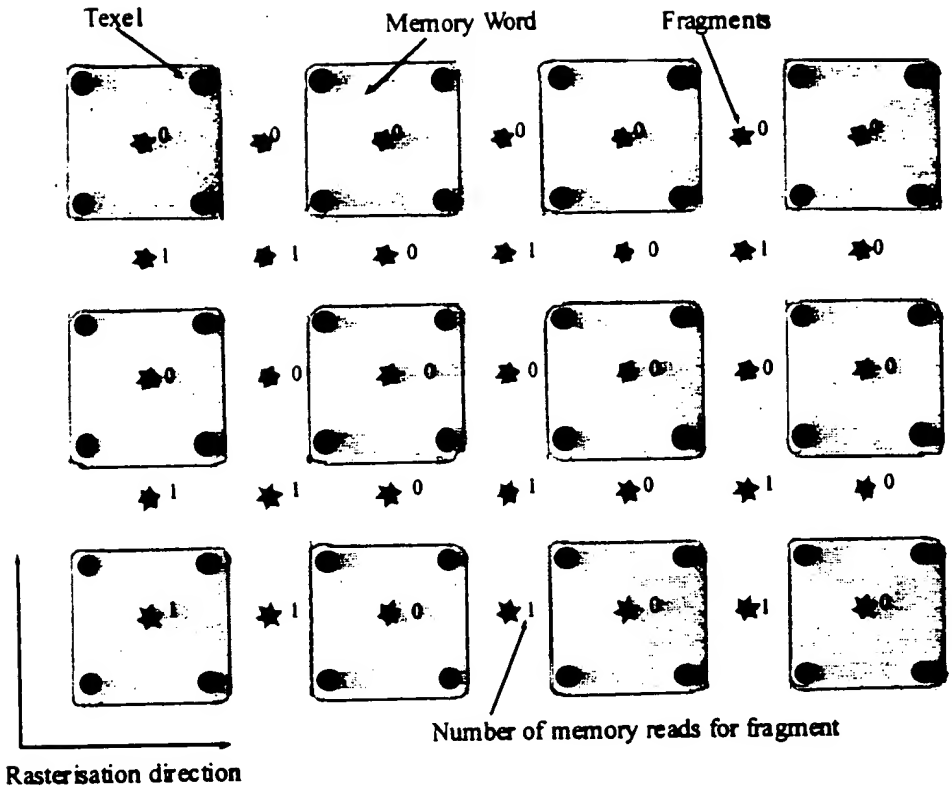

Texel
Memory Word
Fragments
Rasterisation direction
Number of memory reads for fragment

FIG. 9

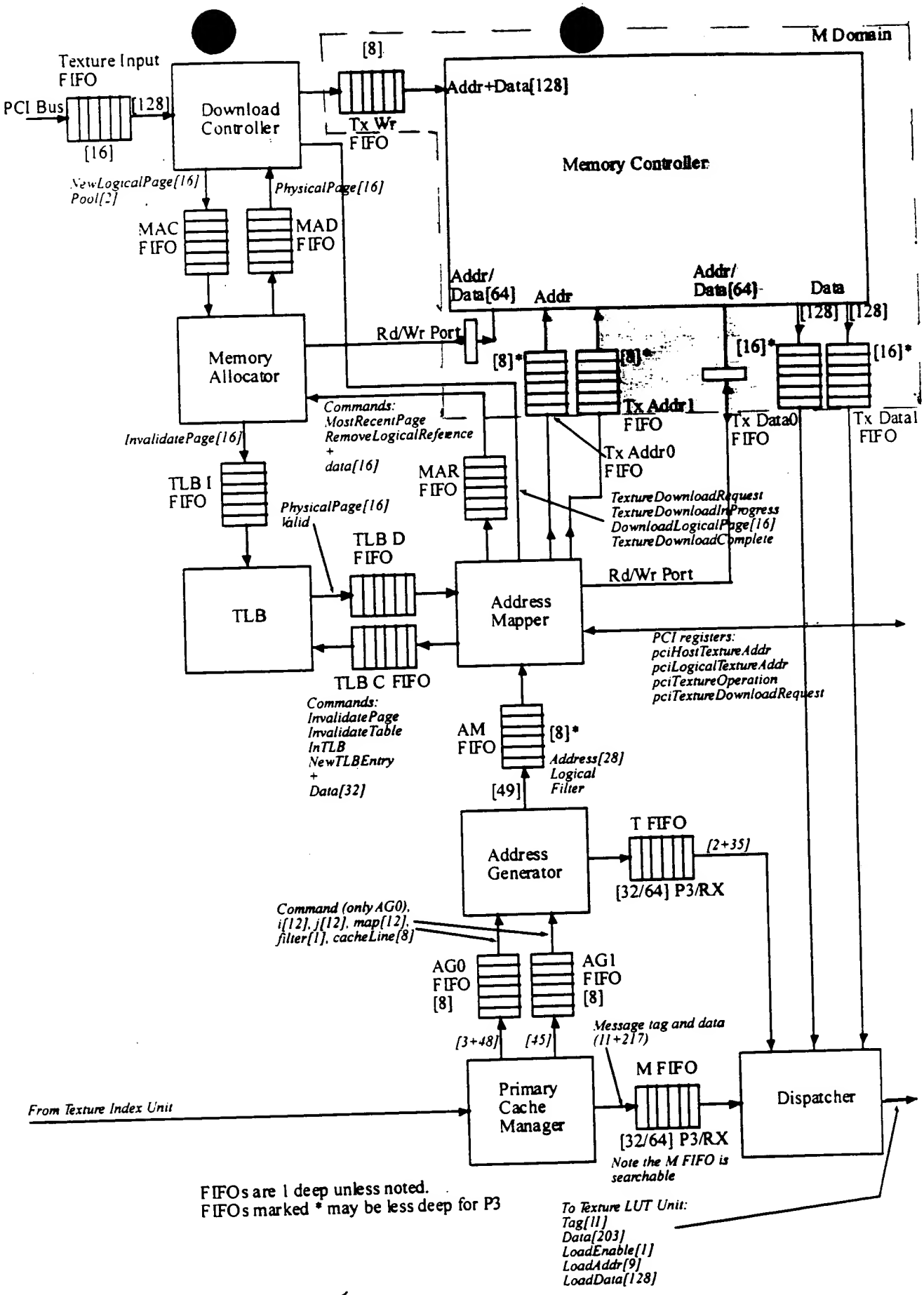

M Domain
Texture Input FIFO
PCI Bus
[128]
[16]
Download Controller
[8]
Tx Wr FIFO
Addr+Data[128]
Memory Controller
NewLogicalPage[16]
Pool[2]
PhysicalPage[16]
MAC FIFO
MAD FIFO
Addr/ Data[64]
Addr
Addr/ Data[64]
Data
[128]
[128]
Memory Allocator
Rd/Wr Port
[8]*
[8]*
[16]*
[16]*
Commands:
MostRecentPage
RemoveLogicalReference
+
data[16]
Tx Addr1 FIFO
Tx Data0 FIFO
Tx Data1 FIFO
InvalidatePage[16]
Tx Addr0 FIFO
TLB I FIFO
MAR FIFO
TextureDownloadRequest
TextureDownloadInProgress
DownloadLogicalPage[16]
TextureDownloadComplete
PhysicalPage[16]
valid
TLB D FIFO
TLB
Address Mapper
Rd/Wr Port
PCI registers:
pciHostTextureAddr
pciLogicalTextureAddr
pciTextureOperation
pciTextureDownloadRequest
TLB C FIFO
Commands:
InvalidatePage
InvalidateTable
InTLB
NewTLBEntry
+
Data[32]
AM FIFO
[8]*
Address[28]
Logical
Filter
[49]
Address Generator
T FIFO
[2+35]
[32/64] P3/RX
Command (only AG0),
i[12], j[12], map[12],
filter[1], cacheLine[8]
AG0 FIFO [8]
AG1 FIFO [8]
[3+48]
[45]
Message tag and data
(11+217)
M FIFO
Primary Cache Manager
Dispatcher
From Texture Index Unit
[32/64] P3/RX
Note the M FIFO is searchable
FIFOs are 1 deep unless noted.
FIFOs marked * may be less deep for P3
To Texture LUT Unit:
Tag[11]
Data[203]
LoadEnable[1]
LoadAddr[9]
LoadData[128]

FIG. 10

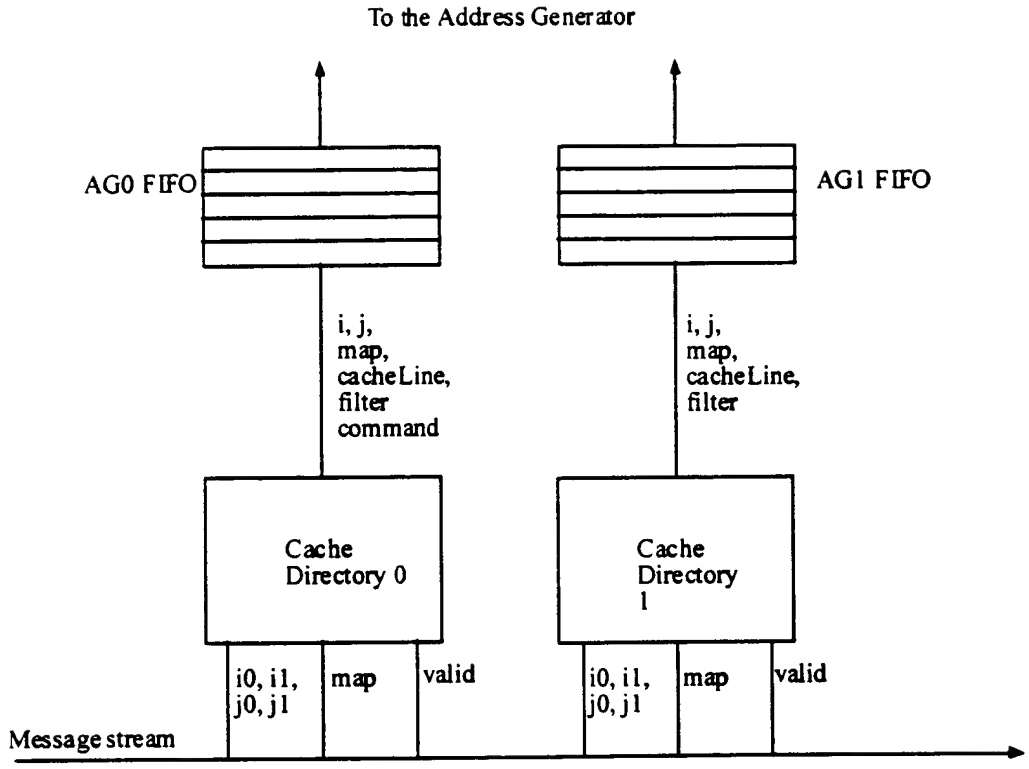

To the Address Generator
AG0 FIFO
AG1 FIFO
i, j,
map,
cacheLine,
filter
command
i, j,
map,
cacheLine,
filter
Cache
Directory 0
Cache
Directory
1
i0, i1,
j0, j1
map
valid
i0, i1,
j0, j1
map
valid
Message stream

FIG. 11

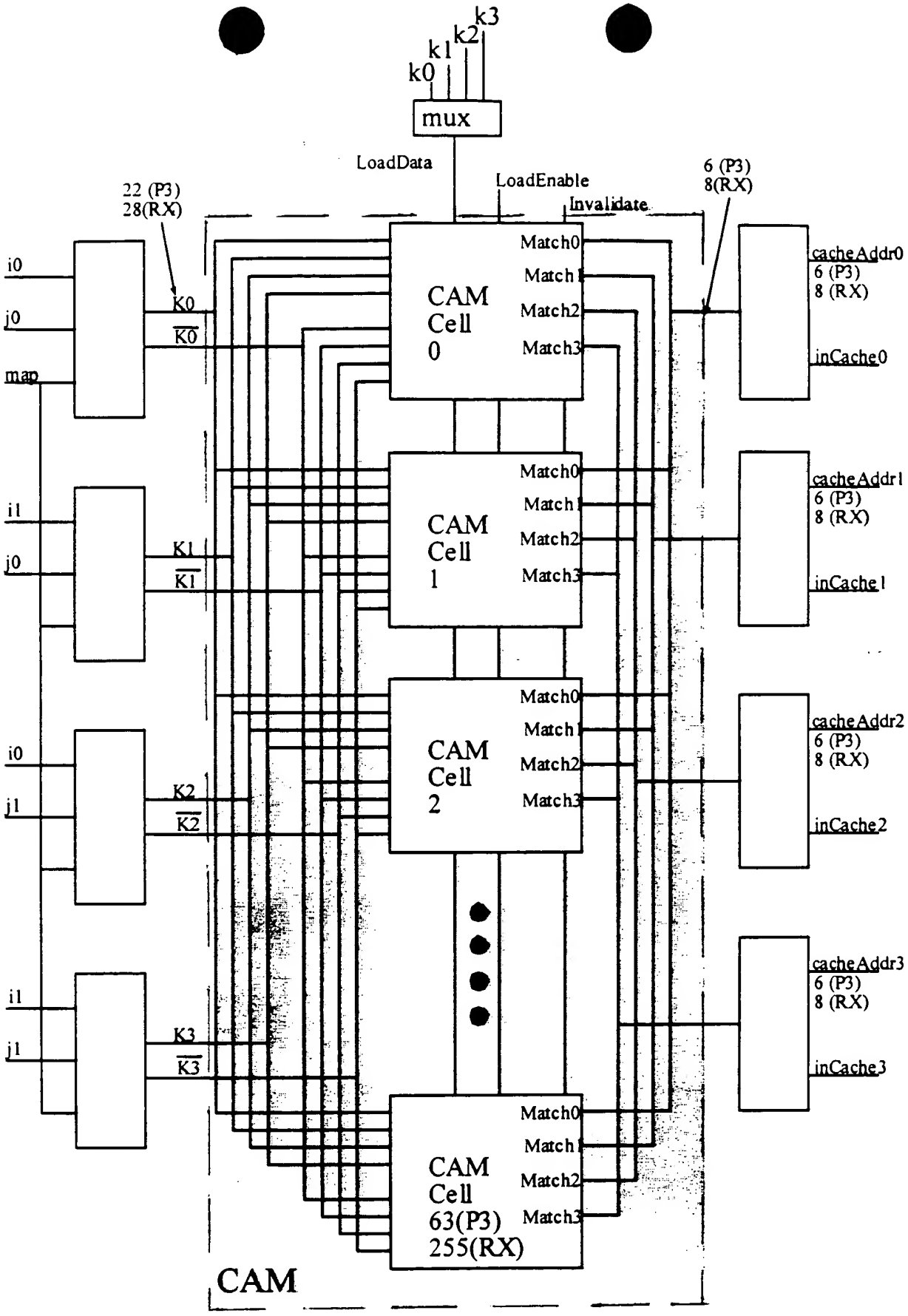

k3
k2
k1
k0
mux
LoadData
LoadEnable
Invalidate
22 (P3)
28(RX)
6 (P3)
8(RX)
i0
j0
map
K0
$\overline{K0}$
CAM
Cell
0
Match0
Match1
Match2
Match3
cacheAddr0
6 (P3)
8 (RX)
inCache0
i1
j0
K1
$\overline{K1}$
CAM
Cell
1
cacheAddr1
inCache1
i0
j1
K2
$\overline{K2}$
CAM
Cell
2
cacheAddr2
inCache2
i1
j1
K3
$\overline{K3}$
CAM
Cell
63(P3)
255(RX)
cacheAddr3
inCache3
CAM

FIG 12

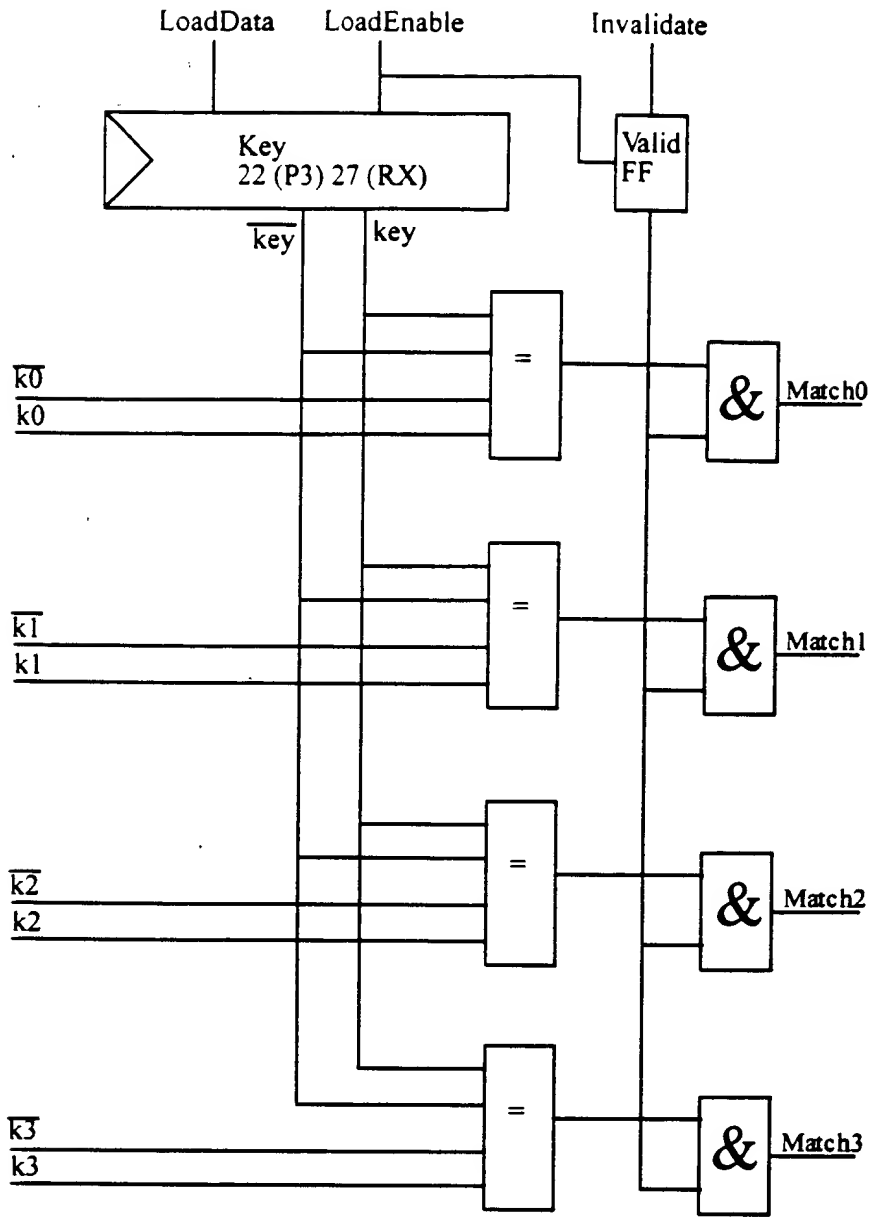

LoadData
LoadEnable
Invalidate
Key
22 (P3) 27 (RX)
Valid
FF
key
key
k0
k0
=
&
Match0
k1
k1
=
&
Match1
k2
k2
=
&
Match2
k3
k3
=
&
Match3

FIG. 13

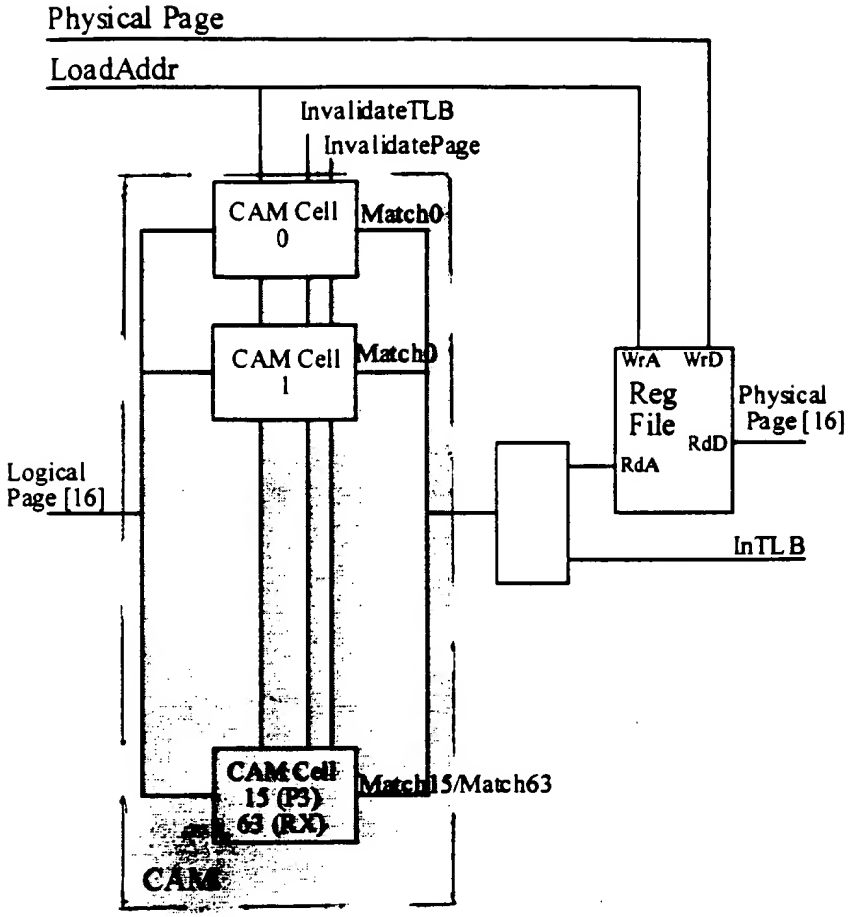

Physical Page
LoadAddr
InvalidateTLB
InvalidatePage
CAM Cell 0
Match0
CAM Cell 1
Match0
WrA
WrD
Reg File
RdD
RdA
Physical Page [16]
Logical Page [16]
InTLB
CAM Cell 15 (P3) 63 (RX)
Match15/Match63
CAM

FIG. 14

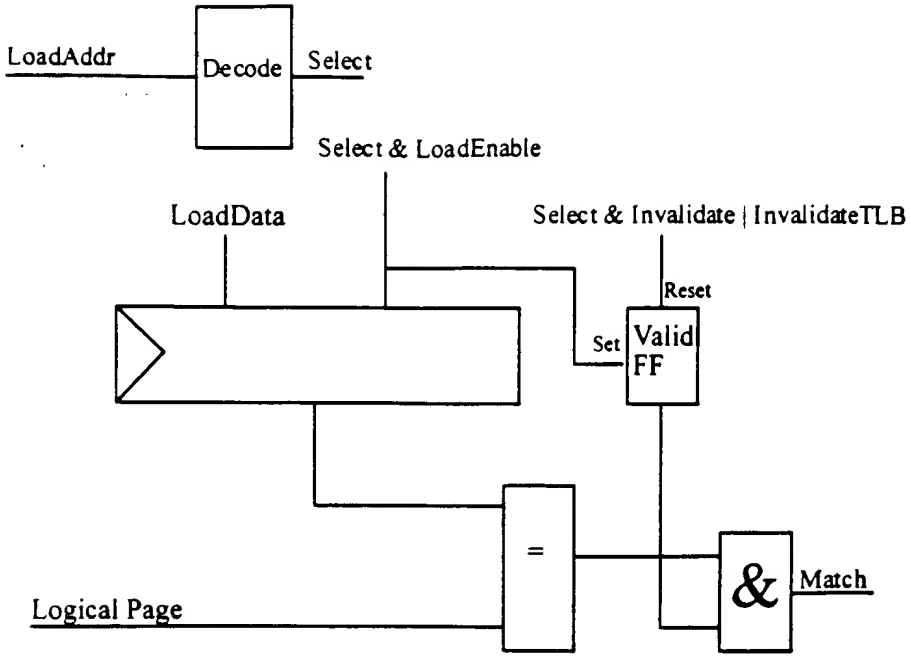

LoadAddr
Decode
Select
Select & LoadEnable
LoadData
Select & Invalidate | InvalidateTLB
Reset
Set
Valid FF
=
&
Match
Logical Page

Fig. 15

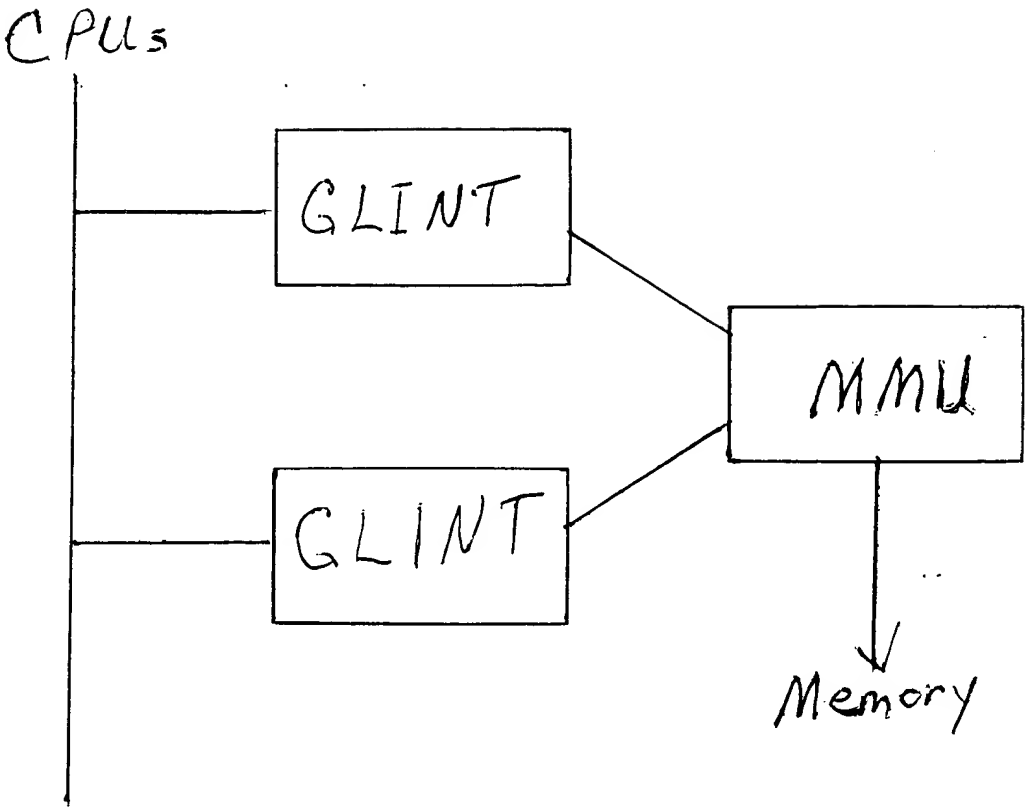


FIG. 16